(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2001－14341
（P2001－14341A）
(43)公開日　平成13年１月19日(2001.1.19)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード（参考）
Ｇ０６Ｆ　17/30		Ｇ０６Ｆ　15/401	３１０Ａ	５Ｂ０７５
		15/40	３７０Ａ	

審査請求　未請求　請求項の数12　ＯＬ　（全 16 頁）

(21)出願番号　特願平11－188613

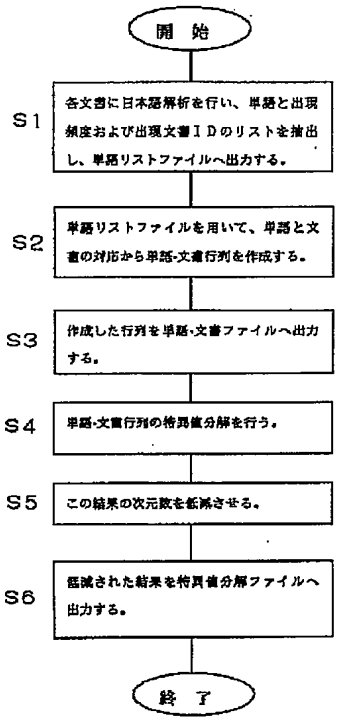
(22)出願日　平成11年７月２日(1999.7.2)

(71)出願人　000006747
株式会社リコー
東京都大田区中馬込１丁目３番６号
(72)発明者　石岡　恒憲
東京都目黒区駒場２－19－23　大学入試センター内
(72)発明者　亀田　雅之
東京都大田区中馬込１丁目３番６号　株式会社リコー内
(74)代理人　100101177
弁理士　柏木　慎史　（外１名）

Ｆターム(参考)　5B075 ND03 NK02 NK32 PQ05 PQ46
PR10 UU40

(54)【発明の名称】　データベース作成装置および関連文書／関連語検索装置、データベース作成方法および関連文書／関連語検索方法ならびに記憶媒体

(57)【要約】
【課題】　小さなメモリ空間でも関連文書／関連語の検索を行うことができるようにする。
【解決手段】　文書群ファイルに含まれた各文書に対して自然言語解析を行い、出現単語、この出現単語の出現頻度、およびその単語が出現した文書のＩＤのリストを抽出し、単語リストファイルに保存する（ステップＳ１）。次に、この単語リストファイル中の出現単語による情報から、行方向に出現単語、列方向に文書番号をとって、単語－文書行列を作成する（ステップＳ２）。そして、単語－文書行列を特異値分解して、特異値ベクトル、単語行列、文書行列を得（ステップＳ４）、この特異値ベクトル、単語行列、文書行列の各々について要素を大きい方からｋ個だけ取り出し、新たな特異値ベクトル、単語行列、文書行列を生成することで次元数を低減する（ステップＳ５）。



1

【特許請求の範囲】

【請求項１】　文書群に出現する単語の抽出を行なって当該単語に関する情報のリストを生成する単語抽出部と、

前記文書群を構成する各文書が前記リストの単語のうちのどれを含んでいるかを示す単語－文書行列を生成する単語－文書対応作成部と、

この単語－文書行列を特異値分解する特異値分解部と、

この特異値分解後の結果から特異値を大きい方から所定数だけ取り出して前記結果を次元低減したデータを作成する次数低減部とを備えている関連文書／関連語検索用のデータベース作成装置。

【請求項２】　前記単語－文書対応作成部は、前記データを用いて関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすいことを修正するように基準化して前記単語－文書行列を生成するものである請求項１に記載のデータベース作成装置。

【請求項３】　請求項１または２に記載のデータベース作成装置で作成された前記データを記憶する記憶部と、

文書検索のための問い合わせと前記文書群の各文書との空間的距離を前記データを用いて測ることにより前記問い合わせに関連がある文書を関連が深い順に前記文書群から検索する関連文書検索部とを備えている関連文書検索装置。

【請求項４】　請求項１または２に記載のデータベース作成装置で作成された前記データを記憶する記憶部と、

文書検索のための問い合わせと前記文書群に含まれる各単語との空間的距離を前記データを用いて測ることにより前記問い合わせに関連がある単語を関連が深い順に前記文書群に含まれる単語から検索する関連語検索部とを備えている関連語検索装置。

【請求項５】　文書群に出現する単語の抽出を行なって当該単語に関する情報のリストを生成する工程と、

前記文書群を構成する各文書が前記リストの単語のうちのどれを含んでいるかを示す単語－文書行列を生成する工程と、

この単語－文書行列を特異値分解する工程と、

この特異値分解後の結果から特異値を大きい方から所定数だけ取り出して前記結果を次元低減したデータを作成する工程とを含んでなる関連文書／関連語検索用のデータベース作成方法。

【請求項６】　前記単語－文書行列生成工程は、前記データを用いて関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすいことを修正するように基準化して前記単語－文書行列を生成するものである請求項５に記載のデータベース作成方法。

【請求項７】　請求項５または６に記載のデータベース作成方法で作成された前記データを用い、文書検索のための問い合わせと前記文書群の各文書との空間的距離を測ることにより前記問い合わせに関連がある文書を関連が深い順に前記文書群から検索する工程を含んでなる関連文書検索方法。

2

【請求項８】　請求項５または６に記載のデータベース作成方法で作成された前記データを用い、文書検索のための問い合わせと前記文書群に含まれる各単語との空間的距離を測ることにより前記問い合わせに関連がある単語を関連が深い順に前記文書群に含まれる単語から検索する工程を含んでなる関連語検索方法。

【請求項９】　文書群に出現する単語の抽出を行なって当該単語に関する情報のリストを生成する工程と、

前記文書群を構成する各文書が前記リストの単語のうちのどれを含んでいるかを示す単語－文書行列を生成する工程と、

この単語－文書行列を特異値分解する工程と、

この特異値分解後の結果から特異値を大きい方から所定数だけ取り出して前記結果を次元低減したデータを作成する工程とにより、関連文書／関連語検索用のデータベースの作成をコンピュータに実行させるプログラムを記憶した、コンピュータに読み取り可能な記憶媒体。

【請求項１０】　前記単語－文書行列生成工程は、前記データを用いて関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすいことを修正するように基準化して前記単語－文書行列を生成するものである請求項９に記載の記憶媒体。

【請求項１１】　請求項９または１０に記載の記憶媒体を用いてコンピュータで作成した前記データを用い、文書検索のための問い合わせと前記文書群の各文書との空間的距離を測ることにより前記問い合わせに関連がある文書を関連が深い順に前記文書群から検索することをコンピュータに実行させるプログラムを記憶した、コンピュータに読み取り可能な記憶媒体。

【請求項１２】　請求項９または１０に記載の記憶媒体を用いてコンピュータで作成した前記データを用い、文書検索のための問い合わせと前記文書群に含まれる各単語との空間的距離を測ることにより前記問い合わせに関連がある単語を関連が深い順に前記文書群に含まれる単語から検索することをコンピュータに実行させるプログラムを記憶した、コンピュータに読み取り可能な記憶媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】この発明は、データベース作成装置および関連文書／関連語検索装置、データベース作成方法および関連文書／関連語検索方法ならびに記憶媒体に関する。

【０００２】

【従来の技術】近年、急速に関心の高まってきているデータマイニングの分野において、その適用分野の一つである文書マイニングは、インターネットのホームページを検索する検索エンジン利用の普及と伴って、コンピュ

ータの非専門家にとってもとりわけ注目の高いところである。

【０００３】文書マイニングでは、扱うデータ量が膨大である上に、実用に耐えうる速度での応答が求められるため、その解決方法の一つとして、我々は単語の共起に基づいた検索アルゴリズムに注目してきた。「単語の共起」とは、同一の文書／文に複数の単語が同時に出現することをいう。

【０００４】従来、単語の共起に着目した「文書マイニング」には大別して２つのアプローチがあったと思われる。一つは、入力キーワードを含む文書集合に成立する相関ルールを求め、そのルールに基づき検索をおこなうものである。発見すべき知識は、どのような単語同士が共起しやすいかである。

【０００５】もう一つのアプローチは、入力キーワード／問い合わせ文と検索対象文書に現われる単語との共起の度合いによって、より適切と考えられる文書を検索するものである。基本的な考え方は、検索要求ベクトルに類似したベクトルをもつ文書は、適切な文書であると判断するもので、一般にはベクトル空間モデル（vector-space model）と呼ばれる。

【０００６】その際に、単語の重み付けがしばしば行なわれるが、その方法として、単一文書中で出現する頻度（within-story term frequency）に応じて重みを与えるtf法と、その単語が出現する文書数の逆数（inverse document frequency）に応じて重みを与える（すなわち、さまざまな文書に出現するありふれた単語の重みを低くする）idf法とを組み合わせたtf・idfモデル、もしくはその派生が用いられることが多い。

【０００７】統計的色彩が強い方法としては、Deerwesterによって提案された“Latent Semantic analysis”がある（“Deerwester, S., Dumais, S. T., G. W. Landauer, T. K. andHarshman, R. (1990):Indexing by latent sematic analysis. Journal of the American Society for Information Science, Vol. 41, No. 7, PP. 391-407”参照）。これは、共起の頻度を示す単語－文書行列を特異値分解（たとえば、“柳井晴夫、竹内啓(1983):射影行列・一般逆行列・特異値分解、UP応用数学選書10、東京大学出版会”参照）することにより、文書の潜在的意味構造を抽出するものである。得られた意味空間において、互いに関連した文書や単語は近接するように構成される。この方法も結果的にはベクトル空間モデルの一つであるが、共起という一種のパターンマッチを間接的に用いているために、「入力キーワードを全く含まないが意味的に近い」文書をも選ぶこともできるようになる。たとえば、“結婚”という語を入力キーワードにして、“結婚”という語を含まないけれども、“披露宴”や“新婚旅行”といった「“結婚”に関連の深い単語を含む文書」を検索することが可能となる。

【０００８】

【発明が解決しようとする課題】しかし、一般に特異値分解は巨大なメモリ空間を必要とし、データ数が数千を越える程度の大きさの問題に対してさえ、計算量の制限からワークステーションやパソコンの性能では実行不可能である。

【０００９】この発明は、小さなメモリ空間でも関連文書／関連語の検索を行うことができるようにすることにある。

【００１０】

【課題を解決するための手段】請求項１に記載の発明は、文書群に出現する単語の抽出を行なって当該単語に関する情報のリストを生成する単語抽出部と、前記文書群を構成する各文書が前記リストの単語のうちのどれを含んでいるかを示す単語－文書行列を生成する単語－文書対応作成部と、この単語－文書行列を特異値分解する特異値分解部と、この特異値分解後の結果から特異値を大きい方から所定数だけ取り出して前記結果を次元低減したデータを作成する次数低減部とを備えている関連文書／関連語検索用のデータベース作成装置である。

【００１１】したがって、単語－文書行列が疎（大半の行列要素が０）であるという点と、特異値分解において特異値の大きい方から限られた個数だけ求めれば関連文書／関連語検索においては十分であるという点とに着目し、特異値分解後の結果から特異値を大きい方から所定数だけ取り出して、この結果を次元低減したデータを作成することにより、作成した関連文書／関連語検索用のデータベースの記憶容量を低減することができる。

【００１２】請求項２に記載の発明は、請求項１に記載のデータベース作成装置において、前記単語－文書対応作成部は、前記データを用いて関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすいことを修正するように基準化して前記単語－文書行列を生成するものである。

【００１３】したがって、関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすい弊害を防止することができる。

【００１４】請求項３に記載の発明は、請求項１または２に記載のデータベース作成装置で作成された前記データを記憶する記憶部と、文書検索のための問い合わせと前記文書群の各文書との空間的距離を前記データを用いて測ることにより前記問い合わせに関連がある文書を関連が深い順に前記文書群から検索する関連文書検索部とを備えている関連文書検索装置である。

【００１５】したがって、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連文書の検索を行うことができる。

【００１６】請求項４に記載の発明は、請求項１または２に記載のデータベース作成装置で作成された前記データを記憶する記憶部と、文書検索のための問い合わせと前記文書群に含まれる各単語との空間的距離を前記デー

タを用いて測ることにより前記問い合わせに関連がある単語を関連が深い順に前記文書群に含まれる単語から検索する関連語検索部とを備えている関連語検索装置である。

【００１７】したがって、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連語の検索を行うことができる。

【００１８】請求項５に記載の発明は、文書群に出現する単語の抽出を行なって当該単語に関する情報のリストを生成する工程と、前記文書群を構成する各文書が前記リストの単語のうちのどれを含んでいるかを示す単語－文書行列を生成する工程と、この単語－文書行列を特異値分解する工程と、この特異値分解後の結果から特異値を大きい方から所定数だけ取り出して前記結果を次元低減したデータを作成する工程とを含んでなる関連文書／関連語検索用のデータベース作成方法である。

【００１９】したがって、単語－文書行列が疎（大半の行列要素が０）であるという点と、特異値分解において特異値の大きい方から限られた個数だけ求めれば関連文書／関連語検索においては十分であるという点とに着目し、特異値分解後の結果から特異値を大きい方から所定数だけ取り出して、この結果を次元低減したデータを作成することにより、作成した関連文書／関連語検索用のデータベースの記憶容量を低減することができる。

【００２０】請求項６に記載の発明は、請求項５に記載のデータベース作成方法において、前記単語－文書行列生成工程は、前記データを用いて関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすいことを修正するように基準化して前記単語－文書行列を生成するものである。

【００２１】したがって、関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすい弊害を防止することができる。

【００２２】請求項７に記載の発明は、請求項５または６に記載のデータベース作成方法で作成された前記データを用い、文書検索のための問い合わせと前記文書群の各文書との空間的距離を測ることにより前記問い合わせに関連がある文書を関連が深い順に前記文書群から検索する工程を含んでなる関連文書検索方法である。

【００２３】したがって、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連文書の検索を行うことができる。

【００２４】請求項８に記載の発明は、請求項５または６に記載のデータベース作成方法で作成された前記データを用い、文書検索のための問い合わせと前記文書群に含まれる各単語との空間的距離を測ることにより前記問い合わせに関連がある単語を関連が深い順に前記文書群に含まれる単語から検索する工程を含んでなる関連語検索方法である。

【００２５】したがって、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連語の検索を行うことができる。

【００２６】請求項９に記載の発明は、文書群に出現する単語の抽出を行なって当該単語に関する情報のリストを生成する工程と、前記文書群を構成する各文書が前記リストの単語のうちのどれを含んでいるかを示す単語－文書行列を生成する工程と、この単語－文書行列を特異値分解する工程と、この特異値分解後の結果から特異値を大きい方から所定数だけ取り出して前記結果を次元低減したデータを作成する工程とにより、関連文書／関連語検索用のデータベースの作成をコンピュータに実行させるプログラムを記憶した、コンピュータに読み取り可能な記憶媒体である。

【００２７】したがって、単語－文書行列が疎（大半の行列要素が０）であるという点と、特異値分解において特異値の大きい方から限られた個数だけ求めれば関連文書／関連語検索においては十分であるという点とに着目し、特異値分解後の結果から特異値を大きい方から所定数だけ取り出して、この結果を次元低減したデータを作成することにより、作成した関連文書／関連語検索用のデータベースの記憶容量を低減することができる。

【００２８】請求項１０に記載の発明は、請求項９に記載の記憶媒体において、前記単語－文書行列生成工程は、前記データを用いて関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすいことを修正するように基準化して前記単語－文書行列を生成するものである。

【００２９】したがって、関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすい弊害を防止することができる。

【００３０】請求項１１に記載の発明は、請求項９または１０に記載の記憶媒体を用いてコンピュータで作成した前記データを用い、文書検索のための問い合わせと前記文書群の各文書との空間的距離を測ることにより前記問い合わせに関連がある文書を関連が深い順に前記文書群から検索することをコンピュータに実行させるプログラムを記憶した、コンピュータに読み取り可能な記憶媒体である。

【００３１】したがって、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連文書の検索を行うことができる。

【００３２】請求項１２に記載の発明は、請求項９または１０に記載の記憶媒体を用いてコンピュータで作成した前記データを用い、文書検索のための問い合わせと前記文書群に含まれる各単語との空間的距離を測ることにより前記問い合わせに関連がある単語を関連が深い順に前記文書群に含まれる単語から検索することをコンピュータに実行させるプログラムを記憶した、コンピュータに読み取り可能な記憶媒体である。

【００３３】したがって、小さな記憶容量の関連文書／

関連語検索用のデータベースを用い、関連語の検索を行うことができる。

【００３４】

【発明の実施の形態】（１）本システムの理論

［特異値分解のもととなるデータ：］単語－文書行列Ｘを行方向に（抽出された）単語のリスト、列方向に各文書を取り、各文書毎に各単語が出現するかどうかを記憶

$$x_{ij}=0 \quad (\text{単語 } i \text{ が文書 } j \text{ に含まれないとき}) \qquad \cdots\cdots (1)$$

とする。

【００３６】［特異値分解：］行列Ｘを以下のように分解する。

$$X=T_0S_0D_0' \qquad \cdots\cdots (2)$$

ここで、T_0はｔ×ｍ行列、S_0はｍ×ｍの正方対角行列（対角要素以外はすべて０）、D_0'はｍ×ｄ行列である。また０≦ｄ≦ｔとする。“´（ダッシュ）”はD_0の転置を示す。S_0の対角要素は大きい順とする。

【００３８】［疎行列（Sparse matrix）に適した特異値分解：］単語－文書行列は一般に巨大な疎行列となる。このような巨大な疎行列に対する特異値問題を解くために、最も単純なアルゴリズムである部分空間反復（サブスペース繰返し）法を用いる。この方法は古典的なベキ乗法（power method）のブロック化と見なすことができる。この部分空間反復法にはいくつかのバリエーションがあるが、その内の一つは、“Ｂ＝Ｘ´Ｘ”に対して次の式を更新する。

【００３９】$Z_i=B^iZ_0$

ここで、“$Z_0=[z_1, z_2, \cdots, z_s]$”は“ｄ×ｓ”である（“ｓ”は、計算上の精度を高めるために必要な特異値の数に余裕を加えた次元数。例えば、必要な特異値の数が５０なら１０を加えて６０とする。以下同じ。）。ｉは、求めるＢの特異値の数ｐまで繰り返す。ここで、列ベクトル“z_j（$1\leqq j\leqq s$）”は、“$\|z_j\|=1$”で、かつ、互いに独立になるように適当に定める。このようにすれば、列ベクトルz_jは行列Ｂの主要な特異値に収束してゆく。これにより、行列Z_iは漸次、

$$\hat{X}\ (X\text{のハット})=TSD'$$

として作成される行列$\hat{X}$は行列Ｘの近似となる。ここで行列Ｔはｔ×ｋ行列、行列Ｓはｋ×ｋの正方対角行列、行列Ｄ´はｋ×ｄ行列である。本発明の実施の形態の対象とする言語データのような場合、経験的にｋは５０～１００程度にするとよい。

【００４２】［関連文書の検索：］ユーザからの問い合わせを擬似文書ｑと考える。例えば、複数個の単語の組み合わせ等から問い合わせ文を作る。この擬似文書ｑから単語を抽出して、ｔ次元の単語ベクトルx_qで表現することができる。これを用いて、文書行列Ｄの行に対応する１×ｋの文書ベクトル

$$d_q=x_q'TS^{-1}$$

ここでd_cは、単語－文書行列Ｘのｃ列を表している、ｔ次元の単語ベクトルx_cを使って“$d_c=x_c'T$

させたものと定義し、次のように表現する。

【００３５】単語－文書行列Ｘは、ｔを単語数、ｄを文書数としたときｔ×ｄ行列で表し、その各要素をx_{ij}としたとき、

$x_{ij}=1$（単語ｉ（$1\leqq i\leqq t$）が文書（$1\leqq j\leqq d$）に含まれるとき）

または

解する。

【００３７】

列ごとの線形独立性を失ってゆく。そこで行列Ｂの大きなｐ個の特異値ペアを近似するために、各ステップにおいて修正Gram-Scmidtプロシージャを用い、z_jを互いに直交にすれば、それらの間の線形独立性が保たれることをBauerは示した（“Bauer, F. L. (1957年): Das Verfahren der Treppeniterationund verwandte zur Losung algebraiscer Eigenwertprobleme, ZAMP, 8, 214-235”参照）。

【００４０】しかしながら、z_jのＢの特異値ベクトルに対する収束速度はわずか１次に過ぎない。そこで部分空間反復に、洗練されたRutishauser（1970年）のritzitプログラム（部分空間反復に、さらにRayleigh-Ritzプロシージャと Chebyshev多項式を経た高速化を行なっている）を使用している（“Rutishauser, H. (1970年): On the rates of convergence of the Lanczos and the block-Lanczos methods, SIAM Jouranal of Numerical Analysis, vol. 17, pp687-706”参照）。

【００４１】［次元低減：］上記行列Ｘの特異値分解で得た行列S_0の対角要素のうち大きいほうからｋ番目までを取り、これを新たな正方対角行列Ｓとする。それに対応して、行列T_0および行列D_0もｋ列までを抜き出し、これを新たな行列ＴおよびＤとする。このとき、

$$\cdots\cdots (3)$$

を導くことができる。

【００４３】ここでＴはｔ×ｋ行列、Ｓはｋ×ｋ正方対角行列である。“´（ダッシュ）”は転置を、“$^{-1}$”は逆行列を示し、“$S=\mathrm{diag}(\mu_1, \mu_2, \cdots, \mu_k)$”としたとき、“$S^{-1}=\mathrm{diag}(1/\mu_1, 1/\mu_2, \cdots, 1/\mu_k)$”である。このとき擬似文書ｑの文書ベクトル$d_q$（ｋ次元ベクトル）に対し、比較の対象とする文書ｃの文書ベクトルをd_c（ｋ次元ベクトル）とすれば、両文書の相関係数ｒ（d_q, d_c）は、両文書がなす角の余弦で与えられる。

【００４４】

$$r(d_q, d_c)=(d_q, d_c)/\|d_q\|\,\|d_c\| \qquad \cdots\cdots (4)$$

S^{-1}”で求められる。これより、擬似文書d_qに近い文書を、近さの順に提示することが可能となる。なお

（４）式の右辺分子の括弧は、内積を示す。

【００４５】［関連語の検索：］擬似文書qは、ｔ次元の単語ベクトルｘqで表現することにより、擬似文書の

$$t_q$$＝（擬似文書ｑに含まれている単語に対応する行列Ｔの行ベクトルt_iの平均）　……　（５）

に定めることができる。このようにすれば、比較の対象とする単語ｃに対する行列Ｔのｃ行で表されるｋ次元ベクトルをt_cとするとき、両単語の相関係数は、（４）式と同じように、

$$r(t_q, t_c) = (t_q, t_c) / \| t_q \| \| t_c \|$$　……　（６）

として与えられる。これより擬似文書qに関係の深い単語をその近さの順で提示することができるようになる。

【００４６】（２）各システムの動作環境

本システムは、システム単体でもネットワーク環境下でも動作する。典型的には、図１に示すように、クライアント２と、データベース作成装置、関連文書／関連語検索装置であるサーバ３とがネットワーク４で接続されているクライアント／サーバシステム１で構成され、データベース５を格納するサーバ３に対して、クライアント２から検索要求を問い合わせ、その検索結果をサーバ３がクライアント２に返すものである。

【００４７】図２は、クライアント２、サーバ３として用いるコンピュータの概略構成を示すブロック図である。図２に示すように、このコンピュータ２，３は、ＣＰＵ６と、ＲＯＭ、ＲＡＭなどの記憶装置７とがバス８で接続されている。また、バス８には、記憶媒体であるＣＤ－ＲＯＭ９を読み取るＣＤ－ＲＯＭドライブ１０を制御するためのＣＤ－ＲＯＭドライブ制御部１１と、ＣＲＴなどの表示装置１２およびキーボード、マウスなどの入力装置１３を制御する入出力制御部１４と、ハードディスク１５を制御するハードディスク制御部１６と、コンピュータ２，３をネットワーク４と接続するためのＬＡＮ制御部１７とが接続されている。クライアント２のハードディスク１５には、ＣＤ－ＲＯＭ９から本システムのクライアント用ソフトがインスツールされ、サーバ３のハードディスク１５には、ＣＤ－ＲＯＭ９から本システムのサーバ用ソフトがインスツールされている。

【００４８】（３）本システムで利用するデータ構造

（Ａ）本システムで使用する各種ファイル

本システムでは、以下の種類のデータ（図３参照）をファイルとしてサーバ３のハードディスク１５へ保持し、検索時に利用する。

・検索対象となる文書群を保持する文書群ファイル２１
・文書群ファイルから抽出した単語を保持する単語リストファイル２２
・文書群ファイルから抽出された単語が各文書で存在するかどうかを示す行列（単語－文書行列Ｘ）を保持する単語－文書ファイル２３
・単語－文書ファイルから特異値分解をした結果を保持する特異値分解ファイル２４

【００４９】（Ｂ）各種ファイルのデータ構造

・文書群ファイル２１（図３（ａ）参照）

座標をその擬似文書が含む単語群の中心（centoroid）、すなわち、

各文書毎に、表題、文書の種類、書誌事項（著者名、出版社名、発行日等）、要約文等から構成される。

【００５０】・単語リストファイル２２（図３（ｂ）参照）

各単語毎に、単語の表記、この単語が文書群中に出現する頻度、および、この単語が出現した文書ＩＤのリストで構成される。

【００５１】・単語－文書ファイル２３（図３（ｃ）参照）

次のような順序で構成する。行列の行数、行列の列数、行列中の非ゼロ要素数、列ごとに読み込んだ場合の各列先頭時点における非ゼロ要素の累積数＋１、各列における非ゼロ要素の行番号、非ゼロ要素の値そのもの（整数、あるいは実数）

【００５２】・特異値分解ファイル２４（図３（ｃ）参照）

このファイルには、次の３つが含まれる。特異値のリスト、特異値分解された単語行列Ｔ、特異値分解された文書行列Ｄ

【００５３】（４）本システムの処理

（Ａ）全体の流れ

本システムの検索を行うためには、先ず、文書群に対して予め文書群で使用している単語の抽出を行い、各文書がこれらの単語のうちどれを含んでいるかを示す単語－文書ファイル２３を作成し、このファイルを特異値分解し、次元の低減を行った結果を特異値分解ファイル２４に保持する。関連文書検索は、その結果を用いて、ユーザからの問い合わせに応じた文書を関連の度合いの大きい順に出力する。また、関連語検索も同様に解析結果を用いて、ユーザーからの問い合わせに応じた関連単語を関連の度合いの大きい順に出力する。

【００５４】（Ｂ）予備データの生成

以下では、図４に示すサーバ３の機能ブロック図、図５に示すサーバ３が行う処理のフローチャートに基づいて、本システムの予備データの生成の処理について説明する。

【００５５】まず、単語抽出部２５において、文書群ファイル２１に含まれた各文書に対して自然言語解析を行い、出現単語、この出現単語の出現頻度、およびその単語が出現した文書のＩＤのリストを抽出し、単語リストファイル２２に保存する（ステップＳ１）。

【００５６】次に、単語－文書対応作成部２６で、この

単語リストファイル２２中の出現単語による情報から、行方向に出現単語、列方向に文書番号をとって、単語－文書行列Ｘを作成し（ステップＳ２）、単語－文書ファイル２３に出力する（ステップＳ３）。

【００５７】尚、大きな文書ほど単語の共起が起きやすいので、１文書に現れる単語数で共起頻度を割り、さらに各文書（各列）における要素の和が１となるように基準化する。この基準化は、関連文書検索において、大きな文書ほど関連文書として検索されやすいことへの対処である。また、基準化の方法としては、各文書（各列）における要素の２乗和が１となるようにしてもよい。

【００５８】下に示した例は、９文書に出現する１２単語の共起関係を現したものである（列方向に文書、行方向に単語をとってあり、[]で示した数字は行および列の番号を示している。）。

【００５９】

【表１】

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.33	0.00	0.00	0.33	0.00	0	0.0	0.00	0.00
[2,]	0.33	0.00	0.25	0.00	0.00	0	0.0	0.00	0.00
[3,]	0.33	0.17	0.00	0.00	0.00	0	0.0	0.00	0.00
[4,]	0.00	0.17	0.25	0.00	0.33	0	0.0	0.00	0.00
[5,]	0.00	0.17	0.25	0.33	0.00	0	0.0	0.00	0.00
[6,]	0.00	0.17	0.00	0.00	0.33	0	0.0	0.00	0.00
[7,]	0.00	0.17	0.00	0.00	0.33	0	0.0	0.00	0.00
[8,]	0.00	0.00	0.25	0.33	0.00	0	0.0	0.00	0.00
[9,]	0.00	0.17	0.00	0.00	0.00	0	0.0	0.00	0.33
[10,]	0.00	0.00	0.00	0.00	0.00	1	0.5	0.33	0.00
[11,]	0.00	0.00	0.00	0.00	0.00	0	0.5	0.33	0.33
[12,]	0.00	0.00	0.00	0.00	0.00	0	0.0	0.33	0.33

【００６０】このような疎行列である単語－文書行列Ｘをそのままハードディスク１５に格納すると膨大な領域を必要とするので、下記のように、Harwell-Boeing sparsematrix fomatで格納することによって（“Duff, et al.(1989年):Sparse MatrixTest problems, ACM TOMS(Transaction on Mathematical Software)Vol.15, No.1, March 1989”参照）、記憶容量の節約、ならびにデータ読み込み時間の大幅な低減をはかることができる。

【００６１】例えば、上述の単語－文書行列Ｘに対しては、以下の形式で単語－文書ファイル２３に格納される（なお、見やすくするために改行を入れて示す）。

```
12 9 28←行列の行数、列数、非ゼロ要素数
1 4 10 14 17 20 21 23 26 29←各列先頭時点における非ゼロ要素の累積数＋１
1 2 3←各列における非ゼロ要素の行番号、ここから
3 4 5 6 7 9
2 4 5 8
1 5 8
4 6 7
10
10 11
10 11 12
9 11 12←各列における非ゼロ要素の行番号、ここまで
0.33 0.33 0.33←非ゼロ要素の値そのもの、ここから
0.17 0.17 0.17 0.17 0.17 0.17
0.25 0.25 0.25 0.25
0.33 0.33 0.33
1
0.5 0.5
0.33 0.33 0.33
0.33 0.33 0.33←非ゼロ要素の値そのもの、ここまで
```

【００６２】次に、特異値分解部２７で、単語－文書行列Ｘを式（２）のように特異値分解する（ステップＳ４）。上記のＸに対して、T_0は次のようになる（S_0は対角要素のみ示す）。

【００６３】

【表２】

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.000	-0.398	0.112	-0.404	0.223	0.146	-0.541	0.253	-0.320
[2,]	0.000	-0.338	0.083	-0.212	0.387	-0.322	0.556	0.074	-0.234
[3,]	-0.001	-0.258	0.034	-0.106	0.584	0.112	-0.023	-0.264	0.524
[4,]	-0.001	-0.397	0.011	0.470	-0.064	-0.220	0.300	0.063	-0.079
[5,]	-0.001	-0.464	0.081	-0.130	-0.447	0.123	0.018	-0.276	0.498
[6,]	-0.001	-0.236	-0.023	0.484	0.096	0.040	-0.265	0.102	-0.021
[7,]	-0.001	-0.236	-0.023	0.484	0.096	0.040	-0.265	0.102	-0.021
[8,]	0.000	-0.383	0.096	-0.221	-0.484	-0.052	0.033	0.101	-0.202
[9,]	-0.039	-0.130	-0.347	0.054	0.040	0.588	0.185	-0.519	-0.455
[10,]	-0.925	0.048	0.349	0.051	0.009	0.122	0.051	0.005	-0.022
[11,]	-0.351	-0.079	-0.675	-0.115	-0.025	-0.537	-0.254	-0.217	0.003
[12,]	-0.140	-0.067	-0.513	-0.075	-0.008	0.380	0.253	0.657	0.257

【００６４】上記のＸに対して、S_0は次のようになる。（S_0は対角要素のみ示す）

```
[1] 1.230 0.783 0.710 0.631 0.488 0.337 0.317 0.24
6 0.136
```

上記のＸに対してD_0は次のようになる。

【００６５】

【表３】

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.000	-0.419	0.107	-0.378	0.808	-0.064	-0.008	0.084	-0.072
[2,]	-0.006	-0.374	-0.064	0.338	0.107	0.345	-0.028	-0.547	0.558
[3,]	-0.001	-0.505	0.096	-0.037	-0.311	-0.350	0.716	-0.038	-0.031
[4,]	0.000	-0.525	0.135	-0.395	-0.479	0.212	-0.511	0.105	-0.059
[5,]	-0.001	-0.366	-0.016	0.752	0.087	-0.137	-0.240	0.358	-0.296
[6,]	-0.752	0.061	0.492	0.082	0.018	0.361	0.160	0.020	-0.160
[7,]	-0.519	-0.020	-0.230	-0.050	-0.017	-0.617	-0.322	-0.431	-0.069
[8,]	-0.380	-0.041	-0.390	-0.072	-0.017	-0.035	0.052	0.596	0.580
[9,]	-0.142	-0.117	-0.714	-0.071	0.005	0.422	0.191	-0.106	-0.474

【００６６】特異値分解によって求められた各行列は、文書数、あるいは単語数が膨大になるとT_0，S_0，D_0を保持するためのメモリ空間も膨大なものとなるので、次数低減部２８で次のようにして次数の低減を図り（ステップＳ５）、その結果を記憶部である特異値分解ファイル２４に出力する（ステップＳ６）。

【００６７】すなわち、特異値ベクトルS_0は、単語－文書行列Ｘのrank（本例では９）の数だけ要素が存在するが、この特異値のベクトルの大きい方からｋ個だけ取り出し、これを新たにＳと置く。これに対応し、行列T_0、D_0に対しても、それぞれ先頭からｋ列を取り出して新たにそれぞれ行列Ｔ，Ｄとおく。このようにして作成されたＴ，Ｓ，Ｄを保持することにより大幅な資源（ディスクやメモリなど）の節約を達成する。例えば、特異値のベクトルの大きい方からｋ＝２個だけ取り出すとすると、行列Ｓは次のようになる（以下では説明の簡便性を考慮してｋ＝２とするが、実際の大規模データに対しては経験的にｋ＝５０～１００程度とすると良い）。

[1] 1.230 0.783

	[,1]	[,2]
[1,]	0.000	-0.419
[2,]	-0.006	-0.374
[3,]	-0.001	-0.505
[4,]	0.000	-0.525
[5,]	-0.001	-0.366
[6,]	-0.752	0.061
[7,]	-0.519	-0.020
[8,]	-0.380	-0.041
[9,]	-0.142	-0.117

【００７１】このようにしたとき、“Ｘ^＝ＴＳＤ´”はＸの近似となる。

上例の行列Ｔは、次のようになる。

【００６８】

【表４】

	[,1]	[,2]
[1,]	0.000	-0.398
[2,]	0.000	-0.338
[3,]	-0.001	-0.258
[4,]	-0.001	-0.397
[5,]	-0.001	-0.464
[6,]	-0.001	-0.236
[7,]	-0.001	-0.236
[8,]	0.000	-0.383
[9,]	-0.039	-0.130
[10,]	-0.925	0.048
[11,]	-0.351	-0.079
[12,]	-0.140	-0.067

【００６９】上例の行列Ｄは、次のようになる。

【００７０】

【表５】

【００７２】

【表６】

15　　16

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]
[1,]	0.131	0.116	0.158	0.164	0.114	-0.019	0.006	0.013	0.036
[2,]	0.111	0.099	0.134	0.139	0.097	-0.016	0.005	0.011	0.031
[3,]	0.085	0.075	0.102	0.106	0.074	-0.011	0.005	0.009	0.024
[4,]	0.130	0.116	0.157	0.163	0.114	-0.018	0.007	0.013	0.036
[5,]	0.152	0.136	0.184	0.191	0.133	-0.021	0.008	0.016	0.043
[6,]	0.077	0.069	0.093	0.097	0.067	-0.010	0.004	0.008	0.022
[7,]	0.077	0.069	0.093	0.097	0.067	-0.010	0.004	0.008	0.022
[8,]	0.126	0.112	0.151	0.157	0.110	-0.018	0.006	0.013	0.035
[9,]	0.043	0.038	0.052	0.054	0.037	0.030	0.027	0.022	0.019
[10,]	-0.015	-0.007	-0.018	-0.019	-0.013	0.858	0.589	0.431	0.158
[11,]	0.026	0.026	0.032	0.033	0.023	0.321	0.225	0.167	0.069
[12,]	0.022	0.021	0.026	0.027	0.019	0.127	0.091	0.068	0.031

【００７３】（Ｃ）関連文書検索

以下では、図６に示すサーバ３の機能ブロック図、図７に示すサーバ３が行う処理のフローチャートに基づいて、本システムの関連文書検索の処理について説明する。

【００７４】クライアント２で入力した問い合わせを、問合わせ入力部２９が受け付け（ステップＳ１１）、受け取った文書検索文（以下、これを擬似文書ｑと呼ぶ）中にある単語（複数のときもある）を、関連文書検索部３０が抽出し、この単語と文書群から抽出された単語リストファイル２２とを照合して、ｔ次元の単語ベクトル

$$d_q = x_q \cdot TS^{-1} = [0, -0.419]$$

次に、このd_qと検索対象である文書群の１つ１つに対して、両文書の相関係数を（４）式で計算する（ステップＳ１４）。ここで各文書の文書ベクトル（文書ｃに対して文書ベクトルをd_cと呼ぶ）は行列Ｄの第ｃ行で与えられるから、例えば、対象文書を文書９とすると、

$d_c = [-0.142\ -0.117]$

となる。これより、文書d_qと文書d_cとの相関係数は、

$r(d_q, d_c) = \{0 \times (-0.142) + (-0.419) \times (-0.117)\} / [\sqrt{\{0^2 + (-0.419)^2\}} \cdot \sqrt{\{(-0.142)^2 + (-0.117)^2\}}] = 0.634$

となる。

【００７８】このようにして、全文書に対する相関係数を求め、相関係数の大きい順に並べ替える。このうちの上位１０番目までの相関係数を選び、この選び出された相関係数に対応する文書を、文書内容とともに、関連文書表示部３１でクライアント２に表示させることができる（ステップＳ１５）。

【００７９】（Ｄ）関連語検索

以下では、図８に示すサーバ３の機能ブロック図、図９に示すサーバ３が行う処理のフローチャートに基づいて、本システムの関連語検索の処理について説明する。

【００８０】クライアント２からユーザが入力した問い合わせを、問合せ入力部２９で受け取る（ステップＳ２１）。この中から単語（複数も可能）を抽出する。このx_qを作成する（要素は基準化しておく）（ステップＳ１２）。

【００７５】例えば、擬似文書ｑが単語1,3を含んでいるとすれば、

$x_q = [0.5,\ 0,\ 0.5,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]$

を作成する。

【００７６】これを用いて、特異値分解ファイルから得た特異値ベクトルＳ、行列Ｔ、Ｄとから擬似文書ｑの文書ベクトルd_qは（７）式のように計算することができる（ステップＳ１３）。

【００７７】

$$\cdots\cdots \quad (7)$$

単語に対する行列Ｔの行ベクトルを関連語検索部３２で取り出す（ステップＳ２２）。そして、この行ベクトルを抽出したすべての単語に対するこの行ベクトルを取り出して平均ベクトルを計算する（（５）式）。例えば、問い合わせに単語1,3を含んでいるとすれば次のようになる。

【００８１】$t_q = [(0.000\ 0.001)/2,\ (-0.398\ -0.258)/2] = [-0.001,\ -0.328]$

次に、比較の対象となるすべての単語の単語ベクトルt_c（ｋ次元ベクトル）に対し、両単語の相関係数を（６）式で計算する（ステップＳ２３）。この単語ベクトルt_cは行列Ｔの各行ベクトルに対応している。例えば、比較の対象とする単語を単語８とすると、

$t_c = [-0.000\ -0.383]$

であるから、単語t_qと単語t_cとの相関係数は次のように計算される。

【００８２】$r(t_q, t_c) = \{(-0.001) \times (-0.000) + (-0.328) \times (-0.383)\} / [\sqrt{\{(-0.001)^2 + (-0.328)^2\}} \cdot \sqrt{\{(-0.000)^2 + (-0.383)^2\}}] = 0.958$

このようにすべての相関係数を計算し、相関係数の大きい順に並び替え、大きい方から相関係数に対応した単語を表示することによって、問い合わせにある単語に関係の深い単語をその近さの順で提示することができる（ス

テップＳ２４）。

【００８３】（５）実際の評価データ

ある全国紙の１年分(１９９４年)の記事から、先頭２，０５５文書と、そこに出現する頻度4以上の４，０４１単語を用いた場合、最もオーソドックスなハウスホルダー変換による特異値分解では、約１２時間（６０×６０×１２＝４３，２００秒）、所要メモリで約２００ＭＢが必要であるが、本システムを用いると（特異値の大きな方から５０個を求めた場合）９．５秒、所要メモリで１２．７ＭＢを要した。

【００８４】また２０，２１１文書と４４，８８３単語における特異値分解では、従来法では（メモリの制約から）計算不能であるが、本システムでは１３．４秒、所要メモリで５３．０ＭＢで計算できた。

【００８５】

【発明の効果】請求項１に記載の発明は、単語－文書行列が疎（大半の行列要素が０）であるという点と、特異値分解において特異値の大きい方から限られた個数だけ求めれば関連文書／関連語検索においては十分であるという点とに着目し、特異値分解後の結果から特異値を大きい方から所定数だけ取り出して、この結果を次元低減したデータを作成することにより、作成した関連文書／関連語検索用のデータベースの記憶容量を低減することができる。

【００８６】請求項２に記載の発明は、請求項１に記載のデータベース作成装置において、関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすい弊害を防止することができる。

【００８７】請求項３に記載の発明は、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連文書の検索を行うことができる。

【００８８】請求項４に記載の発明は、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連語の検索を行うことができる。

【００８９】請求項５に記載の発明は、単語－文書行列が疎（大半の行列要素が０）であるという点と、特異値分解において特異値の大きい方から限られた個数だけ求めれば関連文書／関連語検索においては十分であるという点とに着目し、特異値分解後の結果から特異値を大きい方から所定数だけ取り出して、この結果を次元低減したデータを作成することにより、作成した関連文書／関連語検索用のデータベースの記憶容量を低減することができる。

【００９０】請求項６に記載の発明は、請求項５に記載のデータベース作成方法において、関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすい弊害を防止することができる。

【００９１】請求項７に記載の発明は、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連文書の検索を行うことができる。

【００９２】請求項８に記載の発明は、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連語の検索を行うことができる。

【００９３】請求項９に記載の発明は、単語－文書行列が疎（大半の行列要素が０）であるという点と、特異値分解において特異値の大きい方から限られた個数だけ求めれば関連文書／関連語検索においては十分であるという点とに着目し、特異値分解後の結果から特異値を大きい方から所定数だけ取り出して、この結果を次元低減したデータを作成することにより、作成した関連文書／関連語検索用のデータベースの記憶容量を低減することができる。

【００９４】請求項１０に記載の発明は、請求項９に記載の記憶媒体において、関連文書検索を行う場合に大きな文書ほど関連文書として検索されやすい弊害を防止することができる。

【００９５】請求項１１に記載の発明は、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連文書の検索を行うことができる。

【００９６】請求項１２に記載の発明は、小さな記憶容量の関連文書／関連語検索用のデータベースを用い、関連語の検索を行うことができる。

【図面の簡単な説明】

【図１】この発明の一実施の形態にかかるクライアント／サーバシステムの概略構成を示すブロック図である。

【図２】前記クライアントおよびサーバに用いるコンピュータの構成を説明するブロック図である。

【図３】前記サーバで用いる各種ファイルの構成を説明する図である。

【図４】前記サーバで行う予備データの生成の処理を説明する機能ブロック図である。

【図５】前記サーバで行う予備データの生成の処理を説明するフローチャートである。

【図６】前記サーバで行う関連文書検索の処理を説明する機能ブロック図である。

【図７】前記サーバで行う関連文書検索の処理を説明するフローチャートである。

【図８】前記サーバで行う関連語検索の処理を説明する機能ブロック図である。

【図９】前記サーバで行う関連語検索の処理を説明するフローチャートである。

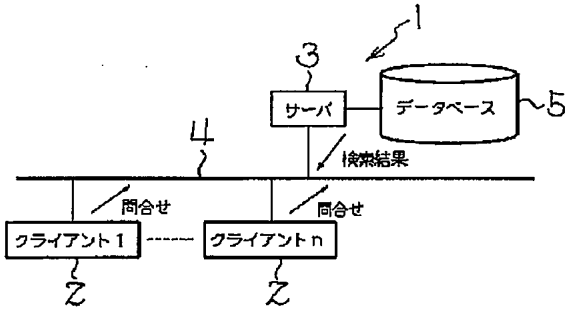
【符号の説明】

３　データベース作成装置、関連文書／関連語検索装置
９　記憶媒体
２４　記憶部
２５　単語抽出部
２６　単語－文書対応作成部
２７　特異値分解部
２８　次数低減部

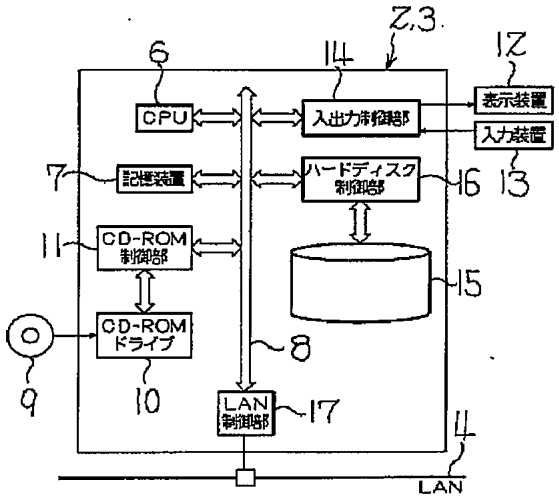
19　　20

３０　関連文書検索部　　３１　関連語検索部

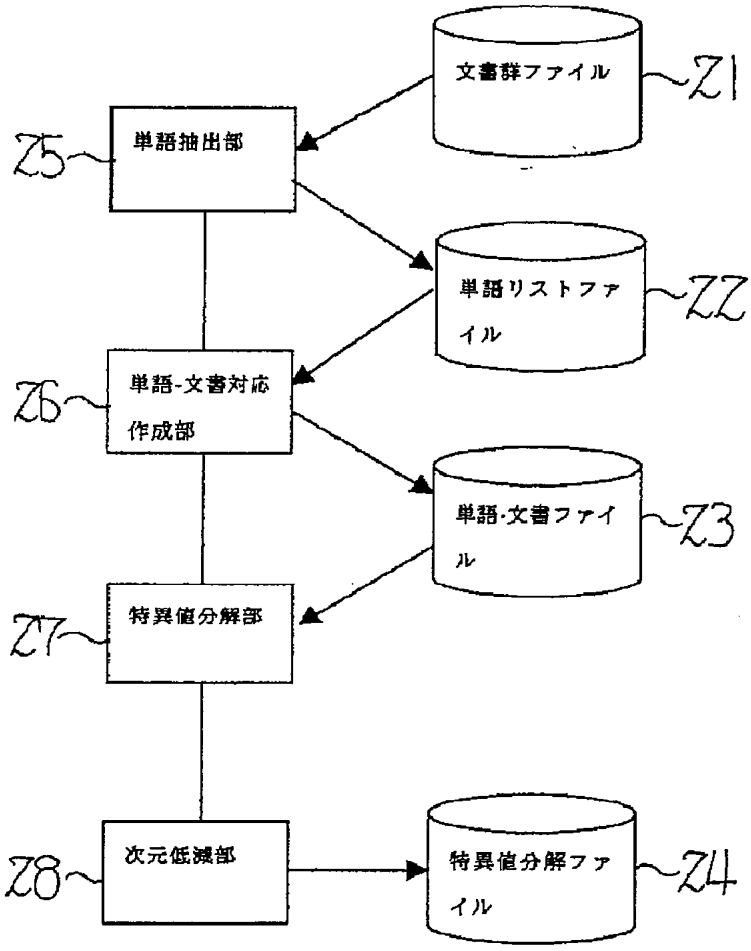
【図１】



【図２】



【図４】



【図３】

(a) 文書群ファイル

Z1

文書ＩＤ	表題	文書の種類	書誌事項	要約文
文書１				
文書２				
：				
：				

(b) 単語リストファイル

Z2

単語ＩＤ	単語表記	出現頻度	出現文書ＩＤリスト	
単語１				
単語２				
：				
：				
：				

(c) 単語-文書ファイル

Z3

A	B	C	D_1	……	D_B	E_1	……	E_B	F_1	……	F_C

A：行列の行数（単語の数ｔ）

B：行列の列数（文書の数ｄ）

C：行列中の非ゼロ要素の数

Ｄｉ：（ｉ－１）列までの非ゼロ要素数の累積＋１

Ｅｉ：ｉ列における非ゼロ要素がある行番号のリスト

Ｆｉ：非ゼロ要素の値（第１列から順に並べる）

Z4

(d) 特異値分解ファイル

λ_1	……	λ_k

T_{11}	……	T_{1k}
：	（行列Ｔ）	：
T_{t1}	……	T_{tk}

D_{11}	……	D_{1k}
：	（行列Ｄ）	：
D_{d1}	……	D_{dk}

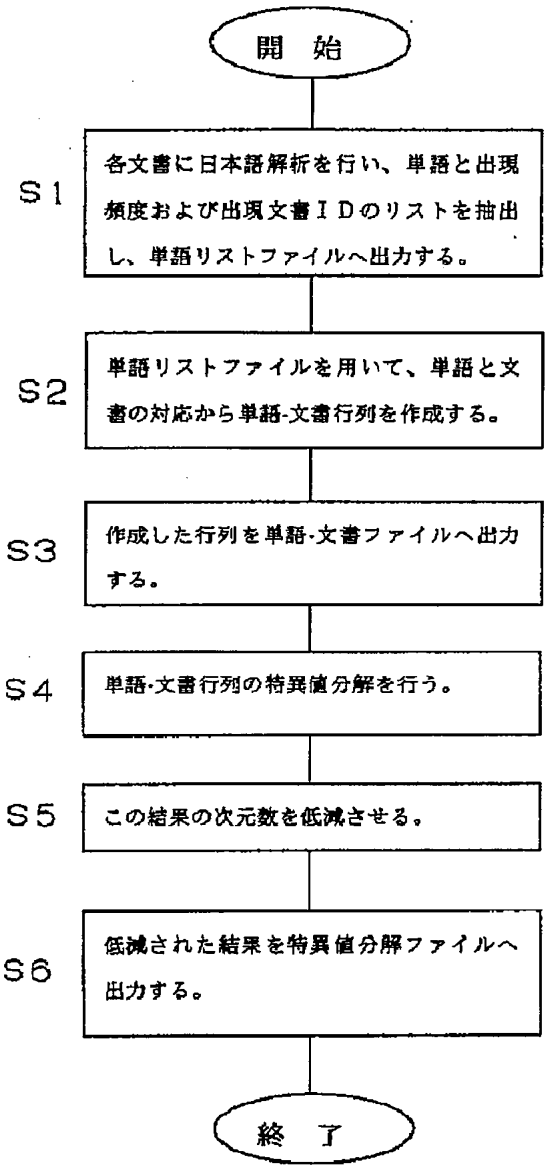
λ_i：ｉ番目の特異値

（ｋは与えられた特異値の数）

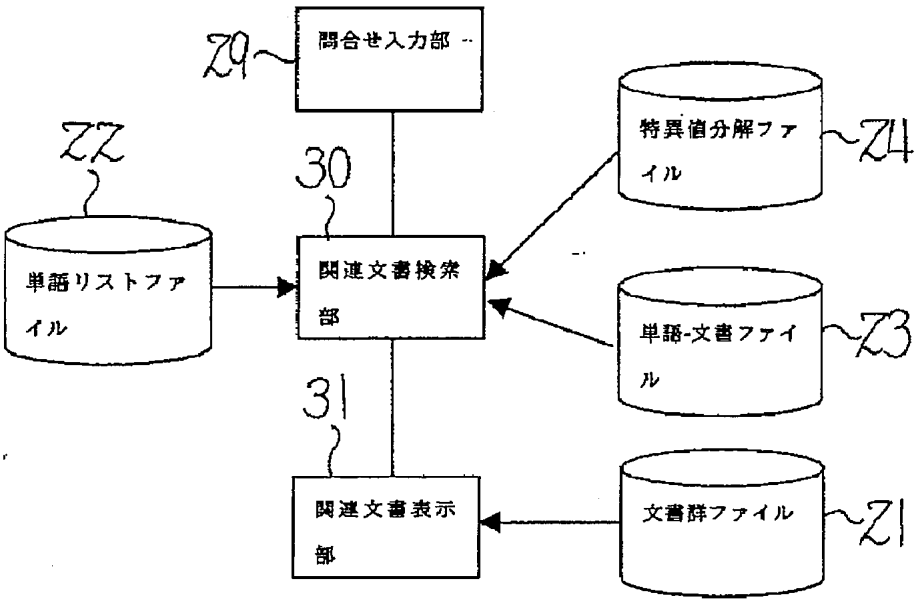
ｔ：単語数

ｄ：文書数

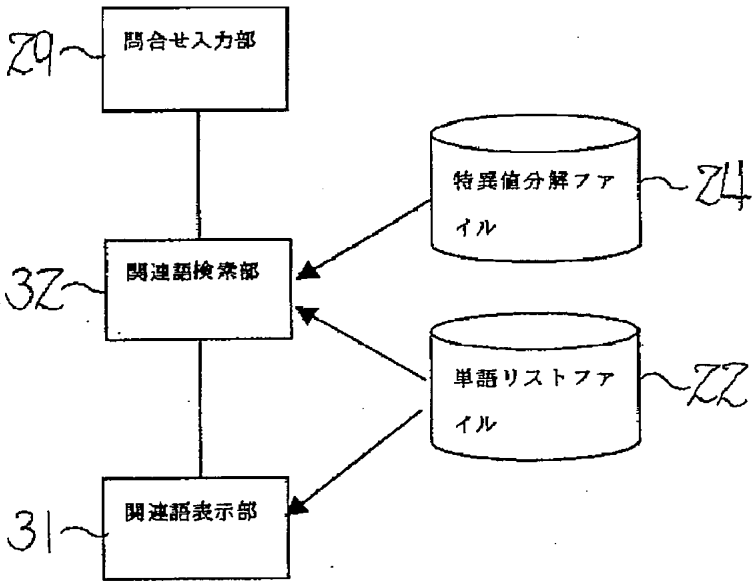
【図５】


開　始
S1
各文書に日本語解析を行い、単語と出現頻度および出現文書ＩＤのリストを抽出し、単語リストファイルへ出力する。
S2
単語リストファイルを用いて、単語と文書の対応から単語-文書行列を作成する。
S3
作成した行列を単語-文書ファイルへ出力する。
S4
単語-文書行列の特異値分解を行う。
S5
この結果の次元数を低減させる。
S6
低減された結果を特異値分解ファイルへ出力する。
終　了

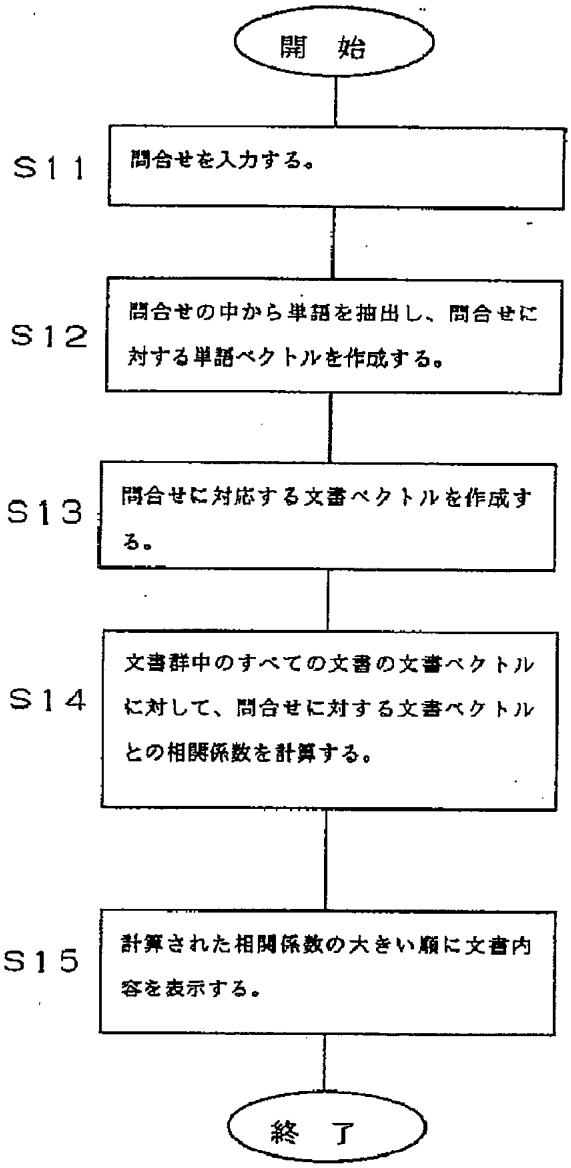
【図６】


29
問合せ入力部
22
30
単語リストファイル
関連文書検索部
特異値分解ファイル
24
単語-文書ファイル
23
31
関連文書表示部
文書群ファイル
21

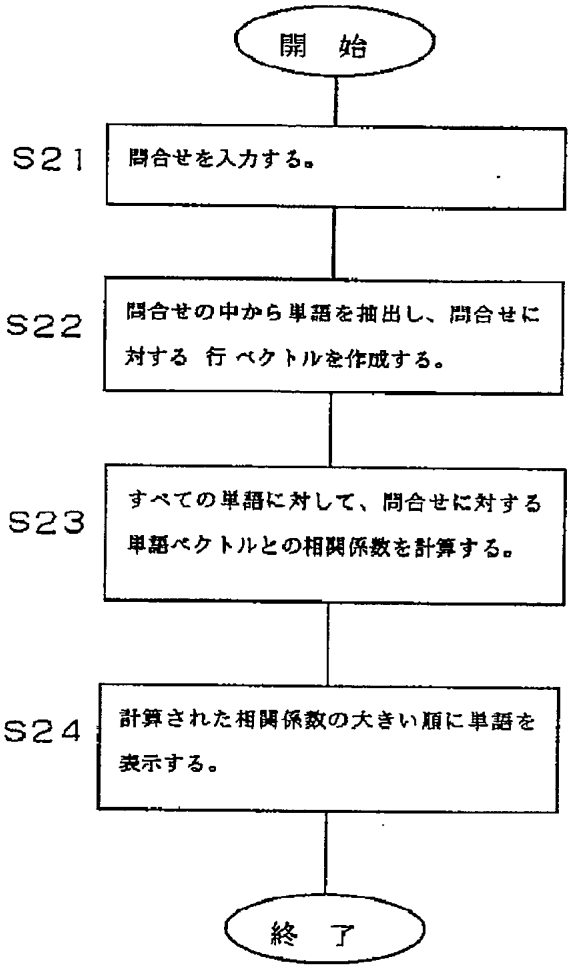
【図８】


29
問合せ入力部
特異値分解ファイル
24
32
関連語検索部
単語リストファイル
22
31
関連語表示部

【図７】


開　始
S11
問合せを入力する。
S12
問合せの中から単語を抽出し、問合せに対する単語ベクトルを作成する。
S13
問合せに対応する文書ベクトルを作成する。
S14
文書群中のすべての文書の文書ベクトルに対して、問合せに対する文書ベクトルとの相関係数を計算する。
S15
計算された相関係数の大きい順に文書内容を表示する。
終　了

【図９】


開　始
S21
問合せを入力する。
S22
問合せの中から単語を抽出し、問合せに対する 行 ベクトルを作成する。
S23
すべての単語に対して、問合せに対する単語ベクトルとの相関係数を計算する。
S24
計算された相関係数の大きい順に単語を表示する。
終　了